

METHOD AND COMPUTER-READABLE MEDIUM FOR DELIVERING HYBRID STATIC AND DYNAMIC CONTENT

5 CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application Serial No. 60/462,300, filed April 14, 2003, which is incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

10 The present invention relates generally to software functional modules and, in particular, to such modules used for delivering hybrid static and dynamic content.

BACKGROUND OF THE INVENTION

Conventionally, access to the internet is provided by way of a browser such as
15 NETSCAPE™ or INTERNET EXPLORER™ installed on a user's computer system. The procedure usually consists of a user opening a browser window and entering the alphanumeric URL (also referred to alternatively as a web page, web site, home page, internet address, etc.). In response, the browser locates the URL on the internet and displays its contents onto the user's screen, sometimes accompanied by audio information.

20 However, this conventional method of internet access may be considered to have various drawbacks, depending on the circumstances. For example, if a user desires to view the contents of two or more arbitrarily chosen Web pages, then a corresponding number of browser windows needs to be opened. Moreover, the desired web page must be entered individually in each window, either by typing or by accessing a directory of bookmarks. In
25 an environment where information needs to be gathered from multiple sources, opening multiple browser windows is inconvenient. Furthermore, some web pages carry multimedia rich content which, if required to be viewed in real time, may consume bandwidth in excess of what is available, resulting in interruptions or degradation of quality. On the other hand, if the multimedia rich content is first downloaded to the user's computer, this may take an
30 excessive amount of time.

Thus, it is apparent that there is a need in the industry for an improved software application which addresses the foregoing problems.

SUMMARY OF THE INVENTION

According to a first broad aspect, the present invention may be summarized as a method of delivering hybrid content to a user of a computer system, comprising instantiating a first functional module on the computer system; commanding instantiation of a second functional module remote from the first functional module; receiving dynamic content from the second functional module; accessing a repository of static content; and delivering hybrid content on the basis of the static content in the repository and the dynamic content received from the second functional module.

The present invention may also be summarized broadly as a computer system, software module or computer-readable storage medium configured and adapted to implement the above method.

According to another broad aspect, the present invention may be summarized as a computer-readable storage medium, comprising a repository of static content as well as a program element for execution by a computing device to deliver hybrid content via a user interface. The program element comprises program code means for commanding instantiation of a functional module remote from the computing device; program code means for receiving dynamic content from the remote functional module; program code means for accessing a repository of static content; and program code means for delivering hybrid content on the basis of the static content in the repository and the dynamic content received from the remote functional module.

For the purposes of this specification, the expression “software module” should be interpreted broadly to include application software, operating system software, hardware drivers, software loaded on a hardware platform, software upgrades and so on, or any combination thereof.

These and other aspects and features of the present invention are apparent to those of ordinary skill in the art upon review of the following description of specific embodiments of the invention in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWING

Fig. 1 shows, in functional block diagram form, a local software module and a remote software module which cooperate to deliver a hybrid static/dynamic content to a user's computer system.

DETAILED DESCRIPTION

With reference to Fig. 1, there is provided a local software module 10 and a remote software module 12 that interact with one another to produce a combined hybrid effect, whereby static content is output as a function of dynamic content. By static content is meant
5 content that is not expected to change in time, such as images, video, audio and so on. On the other hand, dynamic content is content that may change in time and that in fact may be continuously updated, such as the data on the internet at any given time.

The local software module 10 can be designed as an executable or auto-executable application residing on a medium such as a CD-ROM, DVD-ROM, digital audio tape,
10 interactive beacon, etc. In such cases, installation of the local software module 10 on a user's computer system 14 is not required. In other cases, the local software module is downloadable from a network such as the internet. In the example below, it is assumed that the medium on which the first software module is located is a CD-ROM 16. For its part, the remote software module 12 can be designed as a callable application residing on a network
15 server. The remote software module 12 could be programmed in a language such as MACROMEDIA FLASH™, for example.

The local software module 10 and the remote software module 12 communicate via a network 18 such as a local area network (LAN), the public switched telephone network (PSTN), the internet, etc. The functionality of the local and remote software modules 10, 12
20 is now described in greater detail.

A user interested in experiencing a hybrid multimedia presentation inserts a CD-ROM containing the local software module into a drive of the computer system 14. The computer system 14 recognizes the presence of the local software module 10. It should be noted that the computer system 14 is assumed to comprise or have access to a repository of static
25 information 20, such as multimedia content, audio and video streams, databases, bundled MACROMEDIA FLASH™ or DIRECTOR™ applications, etc. In a specific non-limiting embodiment, the local software module 10 could be programmed in VISUAL BASIC™ which encapsulates MACROMEDIA FLASH™ files to allow multimedia functionality, for example.

30 A detection sub-module 22 of the local software module 10 first may be instantiated in order to perform various system checks for ascertaining adequacy of installed hardware, software and network connections. If the detection sub-module 22 indicates that hardware and/or software inadequacies exist, then this can be flagged to the local software module 10,

which takes an action, such as displaying a message on a screen of the computer system 14 or installing the requisite file(s), component(s) or application(s) located on CD-ROM 16.

Upon completion of the detection sub-module 22, assuming that the computer system 14 is equipped with all the necessary hardware, software and network connection, the local software module 10 locates the remote software module 12. This can be done by consulting an internet location (e.g., URL) hard coded in the local software module 10. This internet location can be referred to as a base location.

In any event, the local software module 10 automatically locates the base location without user intervention. Thus, the user is not required to type in the URL of the base location or to click on a particular icon that links to the base location. Stated differently, a user interface 24 “takes the user by the hand”, which may lead to wider acceptance of products embodying the invention.

In some embodiments, the remote software module 12 will be located right at the base location. In variations of this concept, the base location itself might not correspond to the location of the remote software module 12 but might instead only point to the location of the remote software module 12. The address pointed to by the base location would then be used by the local software module 10 to access the remote software module 12. This would allow the location of the remote software module 12 to change over time without having to reprogram the local software module 10.

Once the remote software module 12 has been located, it is instantiated by the local software module 10. Of note is the fact that the remote software module 12 may reside on a server connected to the network 18. The remote software module 12 is adapted to access dynamic content elsewhere on the network 18 and to interact with the local software module 10. The configuration of the remote software module may change over time. Examples of dynamic content include, but are not limited to, HTML, XML, databases, video and audio streams, MACROMEDIA FLASHTM, ACTIONSCRIPTTM, JAVASCRIPTTM, LINGOTM, as well as e-commerce (Security) data, e.g., credit card, user and password information, etc.

The remote software module 12 has the main functionality of controlling the structure and content of the user interface 24, which is implemented by the local software module 10. From the point of view of structure, the remote software module 12 may provide parameters to the local software module 10 such as colors, dimensions, buttons, window sizes, etc. that make up the user interface 24. The local software module 10 receives these parameters from the remote software module 12 and implements the user interface 24 by accessing selected static or dynamic content.

It should be noted that the parameters associated with the user interface 24 may change at some time in the future, and in fact they may vary regularly depending, for example, on the weather, time of day or the day of the week; alternatively, the parameters associated with the user interface 24 may have a dependency on the physical location of the user (as determined through positioning systems), etc. The remote software module 12 will notify the local software module 10 of such changes, resulting in changes to the user interface 24. It is noted that changes to the user interface 24 may thus be made to occur, even after the medium (e.g., CD-ROM 16) has been permanently recorded and distributed to the user.

In addition, the remote software module 12 provides the local software module 10 with dynamic content updates. Based on dynamic content received from the remote software module 12, the local software module delivers selected portions of the static content in the repository 20, in addition to forwarding some of the received dynamic content directly to the user interface 24.

The above mentioned feature is perhaps best illustrated in the context of a concrete, non-limiting example application, of which there are many. To this end, consider a training tool for financial analysts employed by a financial services company. The training tool is distributed in the form of CD-ROMs to new financial analysts or in the context of a continuing education effort. The tool includes a local software module 10 provided on the CD-ROM 16. The local software module 10 is booted automatically upon insertion of the CD-ROM 16 into the CD-ROM drive of a computer system 14. The local software module 10 is responsible for implementing a user interface 24 on the basis of dynamic content received from a remote software module 12 and static content stored in a repository 20 or on the CD-ROM 16 itself. By way of example, the static content stored on the CD-ROM may include a movie of the president's message, images of industry logos and products, databases of statistics and actuarial charts, tests, simulation runs, maps, etc.

The remote software module 12 may be located on the company server, which provides dynamic content to the local software module 10. This dynamic content may be used to control the structure of the user interface 24, as well as content that is specific to the financial analyst training application at hand. For example, the latter type of information may consist of news clips and stock trading statistics from various web sites and a calendar of the day's events. It will be appreciated that such dynamic content provided by the remote software module 12 can have an effect on the static content being accessed and displayed by the local software module 10.

Consider the case where the most actively traded stock is being sought. This type of content is clearly dynamic. If a window in the user interface 24 has been reserved for displaying corporate information (e.g., a video, logo, etc.) regarding the company whose stock is the most highly traded, then the content of such corporate information, which resides
5 in the repository 20, would be extracted by the local software module 10 in order to be displayed in the window 24. This is an example of how high-quality application-specific seemingly dynamic graphical content from various sources can be delivered without heavy usage of the internet. This type of content can thus be referred to as an example of “hybrid” content.

10 Myriad applications for delivering hybrid content have heretofore been unimaginable and hence create their own needs as they arise. For example, consider the case where a club owner is desirous of attracting clientele to his or her club. The club owner distributes CD-ROMs to patrons, each CD-ROM 16 containing its own local software module 10. By accessing the CD-ROM 16 via the computer system 14 at his or her home or work, a patron is
15 presented a user interface 24 which has rich multimedia content, such as music representative of the club’s selection, while providing the patron with access to updated information about new shows at the club.

Also, the possibilities for advertising using hybrid content delivery are tremendous, as advertisers may combine both static multimedia content (such as product photos, video clips,
20 etc.) and light internet content (such as the event being promoted or a current sale) to give the appearance of delivering full-fledged dynamic content from various internet sources. For example, in the club owner scenario above, an advertiser could be provided with its own window in the user interface 24 that cannot be modified by the patron. In addition, the advertisers themselves could be changed through suitable programming of the remote
25 software module 12.

Moreover, the potential for hybrid content delivery in e-commerce applications is immense. In terms of security, control of what is displayed via the user interface 24 is provided by an external entity (the remote software module 12), which can be more easily controlled. In terms of usability, CD-ROMs containing entire product catalogs (repositories
30 of static information 20) could be provided to users. The remote software module 12 in such instances could consist of an application that runs on the company server and interacts with the user through the local software module 10 and the user interface 24. In this way, the user can browse through products by browsing the CD-ROM without having to browse through

the internet; moreover, browsing need not cease when credit card and other information is to be exchanged with the corporation.

Yet another application of the present invention is in the DVD industry, where DVD movie rental typically involves a customer renting a DVD and playing it in a home DVD
5 player. In accordance with an embodiment of the present invention, a DVD will include the local software module 10 as well as a repository of static content 20. The repository of static content 20 includes the movie of interest as well as sounds and images which are common to the beginning of multiple movies. A remote software module 12 could be located on a server belonging to the movie production or distribution company, for example. As increasing
10 numbers of DVD players become integrated with internet access, it will become feasible for movie production or distribution companies to feed dynamic content from the internet to the viewer of the DVD. This can include the latest previews in the form of dynamic content, advertising or news content from various sources. This dynamic content is fed to the local software module, allowing the user to view continuously updatable content in addition to the
15 movie on DVD.

Various enhancements could also be provided. For example, safety features may be built into the captive environment interface 24, such as, by way of non-limiting example, deactivation of one or more functionalities such as mouse buttons or disabling all keys (except, say, for “ALT+TAB” or “CTRL+ALT+DEL”). This may lead to even wider
20 acceptance of the product by a younger audience and particularly their parents or teachers who are often responsible for the computer systems being used by children.

Those skilled in the art should also appreciate that the first and second software modules can be compiled from a high level program written in a number of programming languages for use with many computer architectures or operating systems. For example, the
25 high level program may be written in assembly language, while other versions may be written in a programming language such as C++, JAVA™, VISUAL BASIC™, etc.

While specific embodiments of the present invention have been described and illustrated, it will be apparent to those skilled in the art that numerous modifications and variations can be made without departing from the scope of the invention as defined in the
30 appended claims.